EUROTRONIC
Technology GmbH

Serial Communication Description

Radiator

Stella R Lora

And

Stella Pro Lora

Tabel of Content

1 GENERAL

1.1 REVISIONSHISTORY

Version	Date	Autor	Changes	Reason
1.0	04-02-2025	J. Anders		
1.1	27-03-2025	J. Anders	Description of new commands.	Payload parser changed
1.2	06-05-2025	J. Anders	Add new sections: a) Get radio software b) Get base software c) Wakeup interval	Payload parser changed because of new software version
1.3	02-06-2025	J. Anders	Content correction	Spelling and wording issues

1.2 REVIEW-HISTORIE

Version	Date	Reviewer	Result
1.0	10-02-2025	U. Patil	OK
1.1	28-03-2025	U. Patil	OK
1.2	07-05-2025	U. Patil	OK
1.3	10-06-2025	U. Patil	OK

2 INTRODUCTION

This technical documentation should help to control the radiator Stella R LoRa. The radiator considering the following specifications:

Property	Attribute
LoRa protocol version	1.0.4
Device class	Class A
Communication port	2
FUOTA Port	201
Port to change device class (for FUOTA)	3

2.1 THERMOSTA SOFTWARE

The thermostat consists of a radio module and a base module. The latest software version of the radio software is 0.4.20 and the latest base software version is 1.43.

2.2 PAYLOAD PARSER

The latest version of the payload parser is 1.3.2.

3 TECHNICAL DETAILS

3.1 THERMOSTAT BEHAVIOUR

The thermostat communicates in wakeup intervals. The default wakeup interval is 5 minutes and can be changed from 30 seconds up to 120 minutes. Event-based communication will not be supported.

3.2 COMMAND STRUCTURE

There are two types of command: get commands and set commands. Every command defines a command ID. To use a get command only the command ID will be used. Set commands consists of the command ID and payload. The following table shows an overview of the supported commands:

Command Description	Command ID	Payload required
Set Valve Position	0xFF	yes
Get Valve position	0xFE	no
Set Setpoint Limitation	0xF9	yes
Get Setpoint Limitation	0xF8	no
Set Flags	0xF7	yes
Get Flags	0xF6	no
Set Temperature	0xF5	yes
Get Temperature	0xF4	no
Get Battery Level	0xF3	no
Get Summary	0xE9	no
Set External Temperature Value	0xE4	yes
Get External Temperature Value	0xE3	no
Set max valve position limit	0xE1	yes
Get max valve position limit	0xE2	no
Get Radio software version	0xA0	no
Get Base software version	0xEF	no
Set wakeup interval	0xA3	yes
Get wakeup interval	0xA4	no

In some cases, the parser is sending 0x80. 0x80 means that no changes should be done for a specific payload value.

3.3 COMMAND DESCRIPTION

In this chapter it is described how to control the thermostat with payload parser examples.

3.3.1 Valve Position Control

The valve position sends in terms of 1/100 mm. The range is 0 to 255 (means 2.5 mm). 0 means valve closed at 50 the valve starts opening and 255 means fully open.

3.3.1.1 Set Valve Position

Command	Meaning
0xFF Data 01 Data 02	Data 01: current valve position Data 02: internal use

Payload Parser Example:
{
 "valvePosition":{"currentValvePosition":30}
}
Parser Output: FF 1E 00	

The valve position can only be set as absolute value.

3.3.1.2 Get Valve Position

Command	Response
0xFE	0xFE Data 01 Data 02
Payload Parser Example: {"getValvePosition”:””}	

The following table shows the structure of the get valve position payload:

Data byte	Meaning
Data 01	current valve position
Data 02	Internal use

3.3.2 Setpoint Limitation

The setpoint limitation can be used to configure the minimum and maximum temperature setpoint of the manual temperature regulation. The range of the setpoint limitation is 7.5°C up to 28.5°C and can be changed in 0.5°C steps. If the setpoint limitation is changed, the user cannot set the temperature below the minimum setpoint limitation and above the maximum setpoint limitation on the thermostat. This setting has no effect on the wireless temperature control.

3.3.2.1 Set Setpoint Limitation.

<table>
<tr><th>Command</th><th>Meaning</th></tr>
<tr><td>0xF9 | Data 01 | Data 02</td><td>Data 01: minimum temperature setpoint
Data 02: maximum temperature setpoint</td></tr>
<tr><td colspan="2"><b>Payload Parser Example full package:</b>
{
    "setpointLimitation": {
      "minimumTemperatureSetpoint": 7.5,
      "maximumTemperatureSetpoint": 28.5
    }
 }
Parser Output: F9 0F 39

<b>Payload Parser Example single data:</b>
{
    "setpointLimitation": {
      "maximumTemperatureSetpoint": 28.5
    }
 }
Parser Output: F9 80 39</td></tr>
</table>

The following table shows the structure of the set setpoint limitation payload:

Data byte	Meaning
Data 01	minimum temperature setpoint
Data 02	maximum temperature setpoint

It is not allowed to select a higher minimum temperature setpoint than the maximum temperature setpoint.

3.3.2.2 Get setpoint limitation

Command	Response
0xF8	0xF8 Data 01 Data 02
Payload Parser Example: {"getSetpointLimitation":""}	

The following table shows the structure of the get setpoint limitation payload:

Data byte	Meaning
Data 01	Minimum setpoint limitation
Data 02	Maximum setpoint limitation

3.3.3 Flags

The flags define control structures, status information, and settings. The flags consist of 5 bytes, and every bit of a byte represents an individual state. Every bit can be 0 or 1. 1 means the corresponding setting is enabled or active, and 0 means that the corresponding value is disabled or inactive.

The following tables show the meaning of the flags.

Flag Byte 0

Bits	Meaning	Allowed operation
Bit 7	Internal use	No operation allowed
Bit 6	User changed manual selected temperature	read only
Bit 5	Window timer active	read only
Bit 4	Valve Position Mode	read only
Bit 3	Internal use	No operation allowed
Bit 2	Internal use	No operation allowed
Bit 1	Heating mode	read only
Bit 0	Internal use	No operation allowed

Heating Mode:

The heating mode is the default thermostat mode. In heating mode, the thermostat can be controlled by changing the heating temperature. The thermostat will handle this mode automatically.

Valve Position Mode

In valve position mode, the thermostat can be controlled by changing the heating temperature. The thermostat will handle this mode automatically.

Window timer active

This information shows if the window open detection is active. It is possible to enable the window open detection manually by changing the value to true.

User changed manual selected temperature

This infrastructure shows if user changed temperature manually on the thermostat.

Byte Changed till last reading

This information shows that some flag configurations has been changed since last reading of the flags

Flag Byte 1

Bits	Meaning	Allowed operation
Bit 7	Internal use	No operation allowed
Bit 6	Internal use	No operation allowed
Bit 5	Internal use	No operation allowed
Bit 4	Internal use	No operation allowed
Bit 3	Internal use	No operation allowed
Bit 2	Turn display 180° clockwise	read/write
Bit 1	Advanced Key Lock	read/write
Bit 0	Key Lock	read/write

Key Lock

The key lock is blocking the manual temperature change on the thermostat. It is possible to disable these key lock option on the device. It is not allowed to enable key lock and advanced key lock to same time.

Advanced key lock

The key lock is blocking the manual temperature change on the thermostat. It is not possible to disable these key lock option on the device. The advanced key lock can only be enabled and disabled via command. It is not allowed to enable key lock and advanced key lock to same time.

Turn display 180° clockwise

It is possible to change the display orientation.

Flag Byte 2

Bits	Meaning	Allowed operation
Bit 7	Byte changed 'till last reading	read only
Bit 6	Internal use	No operation allowed
Bit 5	Battery <= 25 %	read only
Bit 4	Battery <= 15 %	read only
Bit 3	Internal use	No operation allowed
Bit 2	Actuator or motor state	read only
Bit 1		
Bit 0		

Actuator or motor state:

The actuator or motor part defines different errors based on the bit configuration. The following table shows the error codes:

Bit Combination	Meaning
000	Actuator inactive
001	Error code E1
010	Reserved
011	Error code E3
100	Reserved
101	Actuator busy
110	In mounting position
111	Device in valve mounting position, wait for button press

3.3.3.1 Set Flags

Command	Meaning
0xF7 | Data 01 | Data 02 | Data 03 | Data 04 | Data 05	Data 01: Flag Byte 0 Data 02: Flag Byte 1 Data 03: Flag Byte 2 Data 04: Flag Byte 3 Data 05: Flag Byte 4

Payload Parser Example:

```
{
   "flags": {
     "keyLock": true,
     "advancedKeyLock": false,
     "displayOrientationChanged": true
 }
  }
```

Parser Output: F7 80 1D 08 80 80

If child protection or display orientation should be changed the whole flag package must be sent. It Is not allowed to enable child protection and child protection plus to the same time.

3.3.3.2 Get Flags

Command	Response
0xF6	0xF6 | Data 01 | Data 02 | Data 03 | Data 04 | Data 05

Payload Parser Example:

{"getFlags":""}

The flag package contains of 5 data bytes. The following table shows the structure of the get flags payload:

Data byte	Meaning
Data 01	Flag Byte 0
Data 02	Flag Byte 1
Data 03	Flag Byte 2
Data 04	Flag Byte 3
Data 05	Flag Byte 4

3.3.4 Temperature

The temperature package consists of the heating temperature, the actual temperature, the temperature offset and window open detection configuration.

Heating Temperature

The heating temperature is the normal setpoint temperature. To change the temperature of the device the heating temperature will be used. The range of the heating temperature is from 7.5°C up to 28.5°C and can be increased or decreased in 0.5° steps. To send the heating temperature the selected temperature must be doubled before sending.

Actual Temperature

The actual temperature is the temperature which is measured from the device itself.

Offset Temperature

The offset temperature will be used only for internal setpoint temperature calculation. The offset temperature defines a range from -5 to +5°C and can be changed in 0.5°C steps.

Window Detection Threshold

The threshold describes the sensitivity of the window open detection. The threshold can be set in 1/12 °C steps but the minimum value must be 4/12 °C. That means that the threshold is defined by three ranges as 4(low), 8(medium), 12(high). By using the value 0xff the window open detection can be disabled.

Window Detection Timer

In case of a detected open window the thermostat will enable the window open detection for a defined time. The time can be 5 up to 30 minutes and be changed in 1-minute steps.

3.3.4.1 Set Temperature

Command	Payload Structure
0xF5 Data 01 Data 02 Data 03 Data 04 Data 05 Data 06 Data 07	Data 01: internal use Data 02: internal use Data 03: heating temperature Data 04: temperature offset Data 05: window detection threshold Data 06: window detection duration Data 07: internal use

Payload Parser Example full package:

```
{
        "temperature": {
          "heatingTemperature": 22.5,
          "temperatureOffset": "0,0",
          "windowDetectionDuration": 10,
          "windowDetectionThreshold": 4
        }
}
```

Parser Output: F580802D00040A80

Payload Parser Example single data:

```
{
        "temperature": {
          "heatingTemperature": 22.5
        }
}
```

Parser Output: F580802D80808080

The parser allows to change single or multiple data.

3.4.4.2 Get Temperature

Command	Response
0xF4	0xF4 Data 01 Data 02 Data 03 Data 04 Data 05 Data 06 Data 07
Payload Parser Example: {“getTemperature”:””}	

The temperature package contains of 7 data bytes. The following table shows the structure of the get temperature payload:

Data Byte	Meaning
Data 01	Actual Temperature
Data 02	Internal use
Data 03	Heating Temperature
Data 04	Offset Temperature
Data 05	Window detection threshold
Data 06	Window detection timer
Data 07	Internal use

3.3.5 Get Battery Level

The battery level is read only and show the current battery level in %. The device is working with battery value between 2.6V and 3.2V.

Command	Response
0xF3	0xF3 Data 01
Payload Parser Example: {"getBatteryState”:””}	

The following table shows the structure of the get battery payload:

Data byte	Meaning
Data 01	Current battery level

3.3.6 Get Summary

The summary command is read only. With summary command the thermostat will send multiple data in one message. The summary contains of the temperature data, some flag bytes, the actual temperature of the device and valve position.

Command	Response
0xE9	0xE9 · Data 01 · Data 02 · Data 03 · Data 04 · Data 05 · Data 06
Payload Parser Example: {“getSummary”:””}	

The following table shows the structure of the get summary payload:

Data byte	Meaning
Data 01	Flag Byte 0 (described in chapter 3.3.3)
Data 02	Flag Byte 1 (described in chapter 3.3.3)
Data 03	Flag Byte 2 (described in chapter 3.3.3)
Data 04	Flag Byte 3 (described in chapter 3.3.3)
Data 05	Actual Temperature of the device (described in chapter 3.3.4)
Data 06	Valve position (described in chapter 3.3.1)

3.3.7 External Temperature Sensor

The Stella thermostats supporting external temperature sensors. The device will take the external temperature sensor values in consideration with internal measured temperature senor value for calculation in internal algorithm and then control heating accordingly. The thermostat can work with external temperatures between 0°C and 50°C. Temperatures allowed in terms of 0.5° steps. Before transmitting the external temperature, the external temperature value will be doubled. Working with external temperature can be disabled by sending 0xFF. The thermostats automatically working with external temperature values after receiving the data.

3.3.7.1 Set external temperature value

Command	Meaning
0xE4 Data 01	Data 01: external temperature value

Payload Parser Example:
{
"externalTemperatureValue":24.0
}
 	
Parser Output: E4 30	

3.3.7.2 Get external temperature value

Command	Respnse
0xE3	0xE3 Data 01
Payload Parser Example: {"getExternalTemperatureValue":""}	

The following table shows the structure of the get summary payload:

Data byte	Meaning
Data 01	The last sent temperature of the external temperature sensor

3.3.8 Max valve position limit

It is possible to set a maximum valve position limit. The thermostat will use the maximum valve position limit to calculate the motor movements instead of the maximum reachable position. The maximum valve position limit will be configured as percentage value between 0 to 100%.

3.3.8.1 Set max valve position limit

Command	Meaning
0xE1 | Data 01	Data 01: max valve position limit in percent

Payload Parser Example:

```
{
        "maxValvePositionLimit": {
          "maxValvePositionLimitInPercent": 90
        }
  }
```

Parser Output: E1 5A

3.3.8.2 Get max valve position limit

Command	Response
0xE2	0xE2 | Data 01

Payload Parser Example:

```
{“getMaxValvePositionLimit”:””}
```

The following table shows the structure of the get max valve position payload:

Data byte	Meaning
Data 01	Max valve position limit in percent

3.3.9 Get radio software version

It is possible to determine the current radio software version.

<table>
<tr><th>Command</th><th colspan="8">Response</th></tr>
<tr><td>0xA0</td><td>0xA0</td><td>Data 01</td><td>Data 02</td><td>Data 03</td><td>Data 04</td><td>Data 05</td><td>Data 06</td><td>Data n</td></tr>
<tr><td colspan="9"><b>Payload Parser Example:</b>
{“getRadioSoftware”:””}</td></tr>
</table>

The following table shows the structure of the get radio software payload:

Data byte	Meaning
Data 01… n	Radio software version number

3.3.10 Get base software version

It is possible to determine the current base software version.

<table>
<tr><th>Command</th><th colspan="8">Response</th></tr>
<tr><td>0xEF</td><td>0xEF</td><td>Data 01</td><td>Data 02</td><td>Data 03</td><td>Data 04</td><td>Data 05</td><td>Data 06</td><td>Data n</td></tr>
<tr><td colspan="9"><b>Payload Parser Example:</b>
{“getBaseSoftware”:””}</td></tr>
</table>

The following table shows the structure of the get base software payload:

Data byte	Meaning
Data 01… n	base software version number

3.3.11 Wakeup Interval

By default, the thermostat works with a wake-up interval of 5 minutes. It is possible to change the wake-up interval from 30 seconds up to 7200 seconds (120 minutes) and can be changed in 1-second steps.

3.3.11.1 Set Wakeup interval

Command	Meaning
0xA3 Data 01 Data 02	Data 01: LSB of calculated wakeup interval Data 02: MSB of calculated wakeup interval

Payload Parser Example:

```
{
  “setWakeupInterval“: {
      "wakeupIntervalInSeconds": 60
  }
}
```

Parser Output: A3 3C 00

3.4.11.2 Get wakeup interval

Command	Response
0xA4	0xA4 Data 01 Data 02

Payload Parser Example:

```
{“getWakeupInterval”:””}
```

The following table shows the structure of the get wakeup interval payload:

Data byte	Meaning
Data 01	LSB Byte of the wakeup interval
Data 02	MSB Byte of the wakeup interval